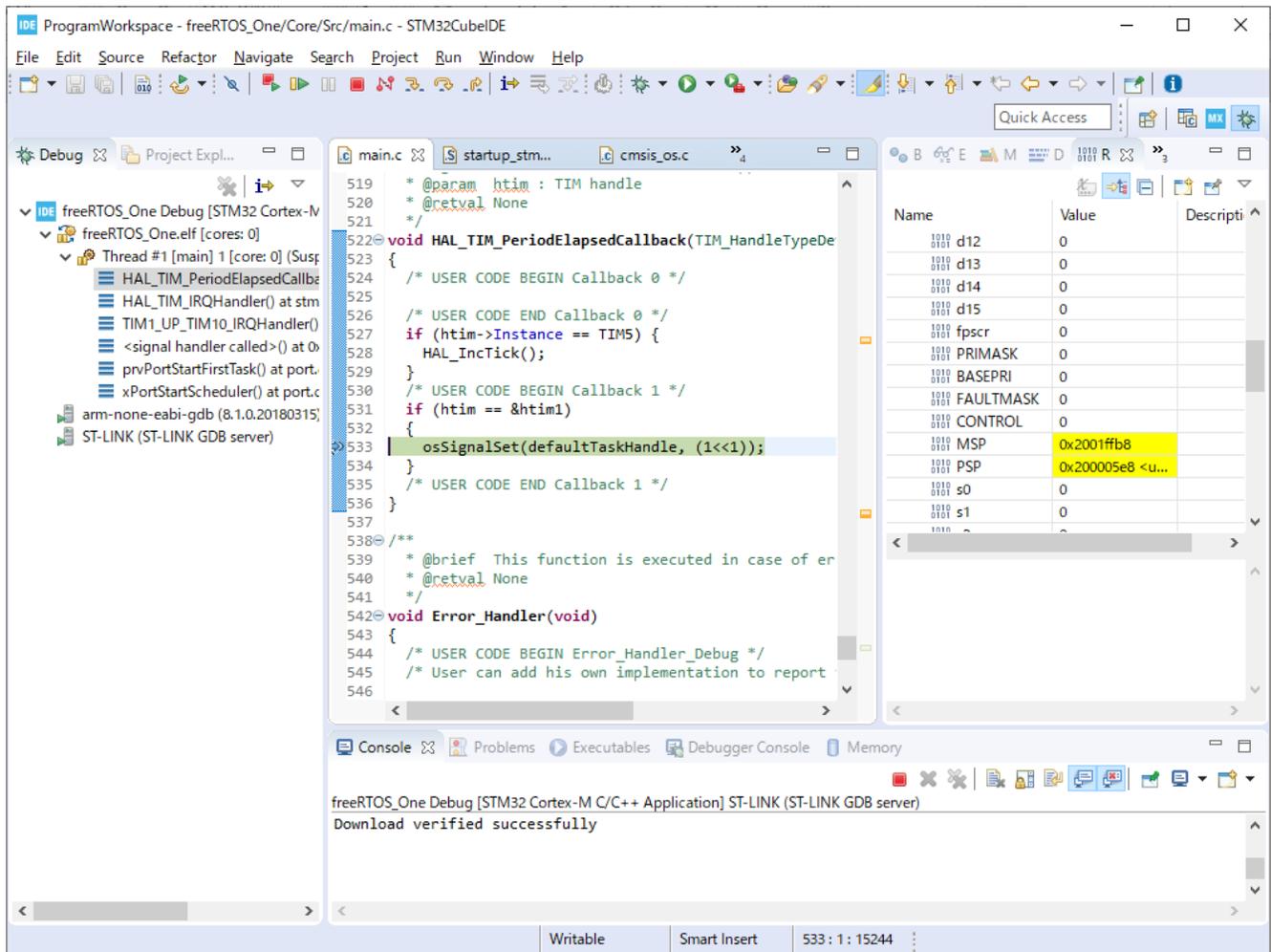


3. 割り込みハンドラ [freeRTOSを解析する] 2020年9月12日

■割り込みハンドラ

TIM1 の割り込みハンドラがどんな状態で飛んでくるか、ブレークをかけて調べてみる。

BASEPRI、PRIMASK、FAULTMASK は全てゼロで、例外は何も禁止されていないことが分かる。
CONTROL の下位 2 ビットが 0 で、特権レベル、メインスタックで動作していることが分かる。
IPSR は 41 で、例外番号 41 の例外処理中であることが分かる。



The screenshot shows the IDE interface with the following components:

- Source Code (main.c):**

```
519 * @param htim : TIM handle
520 * @retval None
521 */
522 void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef
523 {
524     /* USER CODE BEGIN Callback 0 */
525
526     /* USER CODE END Callback 0 */
527     if (htim->Instance == TIM5) {
528         HAL_IncTick();
529     }
530     /* USER CODE BEGIN Callback 1 */
531     if (htim == &htim1)
532     {
533         osSignalSet(defaultTaskHandle, (1<<1));
534     }
535     /* USER CODE END Callback 1 */
536 }
537
538 /**
539 * @brief This function is executed in case of er
540 * @retval None
541 */
542 void Error_Handler(void)
543 {
544     /* USER CODE BEGIN Error_Handler_Debug */
545     /* User can add his own implementation to report
```
- Memory Dump:**

Name	Value	Descripti
0101 d12	0	
0101 d13	0	
0101 d14	0	
0101 d15	0	
0101 fpscr	0	
0101 PRIMASK	0	
0101 BASEPRI	0	
0101 FAULTMASK	0	
0101 CONTROL	0	
0101 MSP	0x2001ffb8	
0101 PSP	0x200005e8 <u...	
0101 s0	0	
0101 s1	0	
- Console:** freeRTOS_One Debug [STM32 Cortex-M C/C++ Application] ST-LINK (ST-LINK GDB server)
Download verified successfully

■osSignalSet